Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Language Identification (LID) in Micro-Texts: An Ultra-Lightweight Geometric Approach.

Oscar Rendón-Aldaraca

oscar.ra@cdvictoria.tecnm.mx

Tecnológico Nacional de México

Héctor Fraire-Huacuja

Tecnológico Nacional de México

Ana María Mendoza-Martínez

Tecnológico Nacional de México

José Ángel Mendoza-Sierra

Tecnológico Nacional de México

Research Article

Keywords: Text classification, Language detection, Vector space model, Euclidean n dimensional space, Character n-grams, Voronoi decomposition, Authorship attribution, Plagiarism detection, Language evolution, Shannon entropy, Multidimensional fields, Text analysis

Posted Date: July 15th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10297975/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Language Identification (LID) in Micro-Texts: An Ultra-Lightweight Geometric Approach.

Rendón-Aldaraca Oscar[1*], Fraire-Huacuja Héctor[2],
Mendoza-Martínez Ana María[3], Mendoza-Sierra José Ángel[1]

[1*]Ciencias Básicas, cdvictoria.tecnm.mx, Boulevard Emilio Portes Gil 1301, Ciudad Victoria, 87010, Tamaulipas, México.
[2]Sistemas, cdmadero.tecnm.mx, Av. 1 de Mayo y Sor Juana Inés de La Cruz, Ciudad Madero, 89440, Tamaulipas, México.
[3]Química, cdmadero.tecnm.mx, Av. 1 de Mayo y Sor Juana Inés de La Cruz, Ciudad Madero, 89440, Tamaulipas, México.

*Corresponding author(s). E-mail(s): oscar.ra@cdvictoria.tecnm.mx;
Contributing authors: automatas2002@yahoo.com.mx;
amendoza_00@yahoo.com.mx; jose.ms@cdvictoria.tecnm.mx;

Abstract

Language identification (LID) in micro-texts (fewer than 10 characters) remains a critical challenge in Natural Language Processing due to severe data sparsity. While contemporary approaches rely heavily on computationally expensive Deep Learning architectures or expansive context windows, this paper introduces an ultra-lightweight geometric alternative. We present a method that encodes text as coordinates within a Real m-Dimensional Euclidean Space (RMDES), where the dimensions represent the bounded, normalized frequency distribution of character unigrams ($m = 27$) and bigrams ($m = 729$). By constraining the feature space to an exact, compact alphabet ($|A| = 27$), the resulting probability vectors maintain mathematical tractability and permit continuous vector space operations. To rigorously test the system's resilience against domain shifts and diachronic variations, the language centroids were calculated using the modern, formal Europarl corpus across 10 distinct languages. Evaluation was subsequently conducted on thousands of randomized, historical text fragments spanning up to 200 years of antiquity extracted from multiple translation matrices of the Bible. The experimental results demonstrate that the proposed Euclidean distance metric on compact bigram spaces successfully overcomes data sparsity, achieving robust classification and outperforming traditional baselines (such as Cavnar & Trenkle) in micro-contexts as short as 5 characters. This method establishes a highly efficient, deterministic framework for language detection, offering significant

computational advantages for edge-computing environments, real-time string sorting, and low-resource linguistic profiling.

Keywords: Text classification · Language detection · Vector space model · Euclidean n dimensional space · Character n-grams · Voronoi decomposition · Authorship attribution · Plagiarism detection · Language evolution · Shannon entropy · Multidimensional fields · Text analysis

1 Introduction

1.1 Importance

Automated text classification is essential for analyzing textual characteristics even before semantic content is fully interpreted. The neutral nature of this analysis makes it broadly applicable in domains such as plagiarism detection, authorship attribution, topic classification, language identification, and language evolution. For instance, educators can use such methods to verify originality in student work, while researchers can analyze large corpora to classify documents based on theme or style. Language detection is particularly important for tasks like machine translation, text retrieval, and computational linguistics.

1.2 Theoretical basis

The Vector Space Model (VSM) traditionally conceptualizes documents within a discrete algebraic framework. While computationally widespread, traditional implementations often treat text features as orthogonal dimensions without exploiting the continuous geometric properties of the underlying space. In contrast, this paper proposes a paradigm shift by encoding character n-gram frequency statistics as coordinate vectors within a real m-dimensional Euclidean space (RMDES), where $m = |A|^n$. For an alphabet A of size $|A| = 27$ (26 letters plus a space symbol), the space bounds are strictly constrained to $m = 27$ for unigrams and $m = 729$ for bigrams. Let a text t be a sequence of symbols belonging to the language universe A^*. The j-th n-gram count within t, denoted as $|g_{n,j}|$, is mapped to a scalar coordinate representing its normalized empirical frequency or probability P_j:

$$P(t) := \left[\frac{c(g_{n,i})}{|g_n|} \right] = [p_i]$$

Although the empirical counts extracted from a finite text string are discrete, the normalized frequencies P_j belong to the set of rational numbers ($\mathbb{Q} \subset \mathbb{R}$). As the sample text size $|t|$ approaches infinity, these rational frequencies act as empirical estimators that can converge toward continuous real values:

$$\lim_{|t|\to\infty} \frac{|g_{n,j}|}{|t|} = P_j \in \mathbb{R}$$

By embedding these statistics into a continuous real vector space $\mathbb{R}^m$, we unlock analytical tools from continuous mathematics—such as basis transformations, gradient fields, and Voronoi decompositions—that are structurally unavailable in purely discrete algebraic models. Furthermore, within a digital environment, each vector component's floating-point

representation is constrained by the machine's architecture. Under standard double-precision specifications (e.g., a 53-bit mantissa), the continuous space is bounded by an information-theoretic capacity. The non-parametric probability of two distinct texts generating the exact same feature vector in an m-dimensional computer space is bounded by:

$$\text{Probability of Collision} = \frac{1}{2^{b \cdot |A|^n}}$$

Where b represents the precision bits of the mantissa. When segmenting a text into a low-dimensional bigram space ($m = 729$), the structural constraints of the language are preserved. This mathematical compression restricts the minimum Shannon entropy of the system, transforming the raw text into a stable topological landscape where distinct languages naturally cluster around geometric barycenters (centroids).

2 Related works

The use of n-grams for text analysis has been explored extensively, since Shannon's foundational work on information theory (Shannon 1948), by Salton, Dubin and others (Dubin 2004; Salton et al. 1975; Wong et al. 1985). Various applications of n-gram based analysis include:

- **Language Identification**: Studies by Cavnar et al., Dunning and Tommi et al, have demonstrated the effectiveness of 1 to 5 character n-grams in distinguishing between languages (Brown 2013; Cavnar and Trenkle 1994; Dunning 1994; Tommi Jauhiainen 2022).
- **Authorship Attribution**: Researchers have employed n-gram analysis for determining authorship by examining stylistic patterns unique to different authors (Basile et al. 2008; Clement and Sharp 2003; Keselj et al. 2003).
- **Plagiarism detection**: Techniques leveraging n-gram frequency distributions have been used to detect similarities between texts and identify cases of potential plagiarism (Jadalla and Elnagar 2007; Lukashenko et al. 2007).
- **Topic Classification and Information Retrieval**: The application of n-grams in text categorization has enabled improvements in search and retrieval tasks (Clement and Sharp 2003; Damashek 1995; Tan et al. 2000).
- **Text Compression and Error Correction**: N-gram models are commonly used in predictive text generation and spell-checking algorithms (Yannakoudakis et al. 1982; Zamora et al. 1981).

Among these methods, classification via n-gram statistics has shown robustness, particularly when used in combination with machine learning models such as Bayesian classifiers, neural networks, and Markov models. Distance metrics, including cosine similarity, Jaccard distance, and entropic distance, have been widely applied to measure text similarity. These methods, often rooted in the Vector Space Model (VSM), rely on term frequency vectors and inner products. While effective, they typically assume orthogonality of features and ignore the geometric properties of the underlying space.

By contrast, the current study introduces a method grounded on a real vector space geometry, using Euclidean distance to measure similarity. This perspective enables a more

flexible and mathematically rigorous framework for text classification, particularly suited for language detection.

3 Proposed Solution

This study proposes a novel classification method based on representing texts as points in a real n-dimensional Euclidean space where each dimension is the frequency of a specific n-gram. The number of dimensions is reduced from a big set of phonemes to 27 symbols from the roman alphabet Σ_{lat} and the space character: $A = \Sigma_{lat} \cup \{'_'\}$ raised to the nth power (Cartesian Product, for n-grams, n is used as power), therefore it is 27 dimensions for Unigrams and 27×27 dimensions for the Bigrams, since there can be 27 symbols for the first character and 27 symbols for the second one. Each unique unigram (or Bigram) has a frequency in any text, and the sum of all the frequencies for any text is 1, thus it is used a floating point number for each dimension. In Perl each floating fraction field has a equivalence to 53 bits of meaning, so the precision for Bigrams is $\frac{1}{2^{53\times2\times27}}$ or $-53 \times 2 \times 27$ in Shannon Information Theory.

The approach follows these key steps:

1. **Encoding Texts as Points**: Each text is transformed into a high-dimensional vector where each component corresponds to the relative frequency of the specific n-gram in the text. So, the vector represents the frequency distribution of the n-grams of a written text. And each dimension of the RMDES represents the probability of a specific n-gram in the text.
2. **Computing Centroids**: The barycenter (centroid) of each pre-classified group of texts is calculated, which constitutes the representative frequencies central point for each class.
3. **Measuring Distances**: The Euclidean distance between an unclassified text and each class centroid is computed. The text is assigned to the class with the nearest centroid.

Mathematically, this method can be described as follows:

1. A set of symbols $\mathbf{A} := \{'a','b','c',...,'z','_'\} = \Sigma_{lat} \cup \{'_'\}$; $|\mathbf{A}| = 27$.
2. $\mathbf{A} \times \mathbf{A} = \mathbf{A}^2 = \{'aa','ab','ac',...,'zz','z_','_a','_b',...,'_z','__'\}$. A set of character Bigrams.
3. $\mathbf{A}^* := \bigcup_{i=1}^{\infty} \mathbf{A}^i$. The set of all the texts that can be represented with 27 close phonemes to symbols of $\mathbf{A}$.
4. t is a text encoded with only the 26 symbols and the space symbol _ of any language, $t \in \mathbf{A}^*$.
5. Let function $\mathbf{g}_n(t) : \mathbf{A}^* \to \{\mathrm{g}_{n,j} \mid \mathrm{g}_{n,j} = t_j; |t_j| = n; \mathrm{g}_{n,j} \in A^n\}$;
   $\mathbf{g}_n(t) = \left[\mathrm{g}_{n,j} = \mathrm{g}_{n,i}, \mathrm{c}(\mathrm{g}_{n,j})\right], \{\mathrm{g}_{n,i}\} \subseteq A^n; j \in \{1,2,...,|t|-n+1\}$ be a function that generates a multiset of n-grams from the simplified phoneme text t.
6. $\mathbf{g}_n(t) := \{(s_i...s_{i+n-1}, \mathrm{c}(s_i...s_{i+n-1}))|1 \leqslant i \leqslant q-n+1\}$, where $q = |t|$ is the size of $t \in \mathbf{A}^q \subset \mathbf{A}^*, t = s_1 s_2 ... s_q$, and $\mathrm{c}(s)$ is the multiplicity of the n-gram s in the multiset $\mathbf{g}$.
   e.g. $\mathbf{g}_2(t) := \{(s_i s_{i+1}, \mathrm{c}(s_i s_{i+1}))|1 \leqslant i \leqslant q-2+1\}, t = s_1 s_2 ... s_q \in \mathbf{A}^q;\ q = |t| \in \mathbb{N}$.
7. Let $\mathbf{P} : \mathbf{A}^* \to \mathbb{R}^m$ be the probability function mapping a text to an m-dimensional space $m = |\mathbf{A}|^n = 27^2$ for 27 symbol Bigrams.
8. $\mathbf{P}(t) := \left[\frac{\mathrm{c}(\mathrm{g}_{n,i})}{|\mathrm{g}_n|}\right] = [p_i]$; $a_i = \mathrm{g}_{n,i}$; $a_i \in \mathbf{A}^n \subseteq \mathbf{g}_n(t)$, where $\mathrm{c}(\mathrm{g}_{n,i})$ is the multiplicity of the element $\mathrm{g}_{n,i} \in \mathbf{g}_n(t)$.

9. The centroid of a class is the vector given by:

$$\mathbf{P}_{\mathrm{c}} = \frac{1}{|\mathrm{C}|} \sum_{t \in \mathrm{C}} \mathbf{P}(t) \tag{1}$$

10. The classification rule assigns a new text to the class with the closest centroid:

$$D(t, \{\mathbf{P}_{\mathrm{c}}\}) = min_{\mathrm{c}} ||\mathbf{P}(t) - \{\mathbf{P}_{\mathrm{c}}\}|| \tag{2}$$

so the time complexity order of the classifier is linear for n different classes: $\mathrm{O(D)} = n$; $\mathrm{O}(n)$.

11. The obtainable Shannon entropy

$$\mathrm{H} = -\sum_{i} \mathrm{P}_i \log_2 \mathrm{P}_i = -\mathbf{P}(t) \log_2 \mathbf{P}(t) \tag{3}$$

,

that is the weighted information quantity in binary form, so the maximum possible information quantity in binary form can be measured as

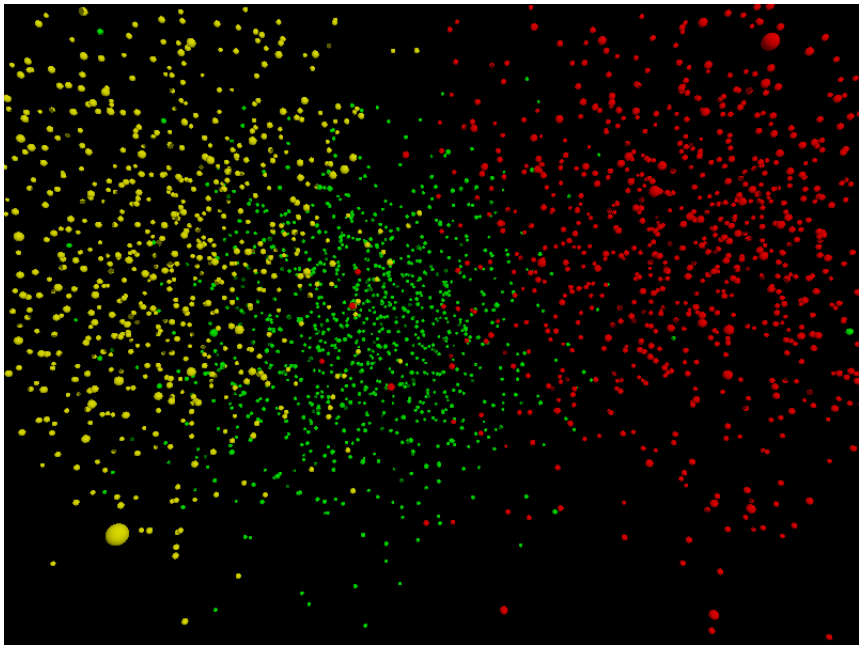
$$\mathrm{I} = \log_2 \mathbf{P}(t) \text{ shannons} \tag{4}$$

,

shannons also known as bits. The maximum information possible in a text t of q symbols is $\mathbf{I}(t) = (8^q)$ shannons, and encoded as floats vector of 53×27^n shannons for n-grams or $53 \times 27^2 = 38637$ shannons (bits) or 2^{38637} different values for the bigrams statistics as allowed by the mantissa of 53 bits for each 2-gram dimension in the Perl programming language.

This method effectively partitions the text space into Voronoi cells, where each pair of closer centroids defines a distinct class boundary as each text is within the bounds of the closer barycenter. The separation is not attenuated in closer distance vectors, so in evolution studies arguments in the similitude does not get distorted, also changing the real entropy of the arguments. The control and certainty of the results can be controlled in a better way as changing vector base, data standard deviation, hyperball radius of the class, principal components, differential, gradient, etc. Furthermore, by treating character n-grams as independent fixed dimensions, the method remains adaptable to various classification tasks beyond language detection as intended by.

Figure 3.1 3 Voronoi cells simulation with random points in R^3.



3.1 Algorithmic Implementation

To operationalize the transition from raw text to the Real m-Dimensional Euclidean Space (RMDES), the classification pipeline is split into two primary phases: Centroid Training and Geometric Inference.Algorithm 1 details the deterministic mapping of a text string into a normalized frequency vector within $\mathbb{R}^{729}$ (for bigrams). Algorithm 2 describes the linear-time classification rule based on the minimum Euclidean distance to the pre-calculated language barycenters.

```
================================================================================
Algorithm 1: Text-to-Vector Mapping via Compact Bigram Counting
================================================================================
Input:  A raw text string 't'
        A bounded alphabet A of size 27 (26 lowercase letters + 1 space)
Output: A normalized probability vector P_t in R^729

1:  Initialize a frequency array V of size 729 with all zeros
2:  Clean text 't': convert to lowercase, replace punctuation/newlines/ with
    spaces, replace accented and special language characters (á,ü,ñ,...) with
    spaces and collapse consecutive spaces into a single space
3:  Let q <- length of cleaned text 't'
4:  If q < 2 Then
```

```
5:      Return V  // Handle exceptionally empty or ultra-short files
6:  End If
7:
8:  // Slide a window of size n=2 across the text
9:  For i <- 1 To q - 1 Do
10:     char1 <- t[i]
11:     char2 <- t[i+1]
12:
13:     // Map the character pair to a unique spatial index in [0, 728]
14:     idx1 <- index_of(char1) // mapped to [0..26]
15:     idx2 <- index_of(char2) // mapped to [0..26]
16:     vector_index <- (idx1 * 27) + idx2
17:
18:     // Increment coordinate count
19:     V[vector_index] <- V[vector_index] + 1
20: End For

21:
22: // Normalize to project the points onto the rational/real hyperplane (Sum=1)
23: Total_Bigrams <- q - 1
24: P_t <- V / Total_Bigrams
25:
26: Return P_t
================================================================================
```

```
================================================================================
Algorithm 2: Minimum Euclidean Distance Inference (Voronoi Classification)
================================================================================
Input:  An unclassified micro-text 't'
        A set of K trained Language Centroids {Pc_1, Pc_2, ..., Pc_K}
Output: Assigned Language Class C_final

1:  // Map the unknown text to its geometric coordinates using Algorithm 1
2:  P_t <- Map_Text_To_Vector(t)
3:
4:  Initialize Min_Distance <- Infinity
5:  Initialize C_final <- Null
6:
7:  // Compute geometric proximity to barycenters
8:  For each Centroid Pc_k in {Pc} Do
9:      Sum_Squares <- 0
10:
11:     // Calculate standard Euclidean distance in m-dimensional space
12:     For i <- 0 To 728 Do
```

```
13:         Difference <- P_t[i] - Pc_k[i]
14:         Sum_Squares <- Sum_Squares + (Difference * Difference)
15:     End For
16:     Distance <- SquareRoot(Sum_Squares)
17:
18:     // Closest barycenter check (Voronoi cell assignment)
19:     If Distance < Min_Distance Then
20:         Min_Distance <- Distance
21:         C_final <- Class_Label(Pc_k)
22:     End If
23: End For
24:
25: Return C_final
================================================================================
```

4 Experimental results

To evaluate the proposed method, 10 groups of text were made, each in one of the languages from the europarl v6 (Koehn 2005a,b): Danish, English, Finish, French, German, Hungarian, Italian, Lithuanian, Slovenian, and Spanish — calculating their class centroids (baricenters) $\mathbf{P}_c(1)$ for each of these languages.

The texts to classify were random takes obtained from 5 versions of the Bible in (Bible-Gateway.com 2025): French (La Bible du Semeur), Italian (Nuova Riveduta 2006), Spanish (Nueva Versión Internacional), German (Hoffnung für Alle) and English (New King James Version).

For each Bible version, 100 random fragments were extracted from the first 10 chapters of Matthew. The fragments were sampled at lengths of 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 characters, resulting in a total of 5,500 generated files. :

$$5500 \text{ files } = \Big(100 \text{ fragments } \times 5 \text{ languages} \times \Big| \{5,10,20,30,40,50,60,70,80,90,100\} \text{ characters } \Big|\Big). \tag{5}$$

As there are 10 language baricenters the identification probability is $p = 0.1$

So the numbers for a random take are:

$$\mathscr{E}_i = \{N = 500; n = 100; p = 0.1\}; i \in \{5,10,20,30,40,50,60,70.80,90,100\} \tag{6}$$

The results are shown in tables 4.1 and 4.2. In the first column we have the length of the test archives in characters, the second column has the fraction of the 100 files in "de" German correctly classified, the third, fourth, fifth and sixth columns show the same but in English, Spanish, French, and Italian; respectively; so 1 represents all correctly classified, 0.5=50/100 correctly classified.

Table 4.1 shows the results using function $\mathbf{g}_1$ to extract Unigrams from text convert them to a vector with the frequency of each Unigram and measuring the distance to 10 centroids

$D(\mathbf{P}(\mathbf{g}_1(t)), \{\mathbf{P}_c\})$ choosing the smallest distance to be the center of the belonging class of each of the 5500 texts.

Table 4.2 shows the results using Bigram function $\mathbf{P}(\mathbf{g}_2(t))$ and the Euclidean distance from the centroids to the 5500 texts.

In Figure 4.1, we can see the mean of the results when using each of the 2 methods and the effectiveness of the methods depending on the length of the text analyzed. The results show that it is a good method for language identification, seems intuitively more effective than untrained humans, because with only 4 letters and a space (or 5 letters), the program could identify the language choosing from 10 languages. The best results were obtained when using Bigrams statistics , with the proposed Euclidean distance, the same as cosine similitude $D(\mathbf{P}(\mathbf{g}_2(t)), \{\mathbf{P}_c\})$ (Table 4.2).

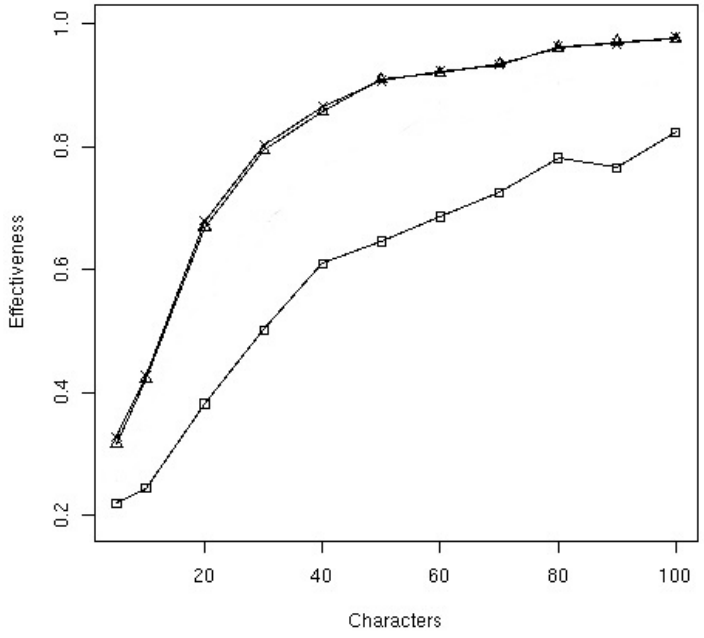
Table 4.1 Efectivity detecting language using 5 to 100 chars texts segments using Unigrams $D(\mathbf{P}(\mathbf{g}_1(t)), \{\mathbf{P}_c\})$ in 10 different languages and 11 text lengths

"Chars"	"de"	"en"	"es"	"fr"	"it"
"100"	"0.88"	"0.85"	"0.86"	"0.86"	"0.67"
"90"	"0.83"	"0.75"	"0.81"	"0.82"	"0.62"
"80"	"0.85"	"0.79"	"0.79"	"0.8"	"0.68"
"70"	"0.84"	"0.76"	"0.84"	"0.65"	"0.54"
"60"	"0.8"	"0.74"	"0.6"	"0.75"	"0.54"
"50"	"0.79"	"0.65"	"0.64"	"0.67"	"0.48"
"40"	"0.74"	"0.62"	"0.64"	"0.58"	"0.47"
"30"	"0.67"	"0.47"	"0.49"	"0.52"	"0.36"
"20"	"0.52"	"0.39"	"0.36"	"0.31"	"0.33"
"10"	"0.36"	"0.29"	"0.16"	"0.25"	"0.16"
"5"	"0.42"	"0.23"	"0.17"	"0.18"	"0.1"

Table 4.2 Efectivity detecting language using from 5 to 100 chars text segments using Bigrams distance $D(\mathbf{P}(\mathbf{g}_2(t), \{\mathbf{P}_c\})$, in the 10 different languages and 11 text lengths

"Chars"	"de"	"en"	"es"	"fr"	"it"
"100"	"0.97"	"0.98"	"0.97"	"0.96"	"1"
"90"	"0.97"	"0.99"	"0.97"	"0.94"	"0.99"
"80"	"0.94"	"0.98"	"0.94"	"0.97"	"0.98"
"70"	"0.95"	"0.97"	"0.93"	"0.86"	"0.97"
"60"	"0.93"	"0.96"	"0.85"	"0.89"	"0.97"
"50"	"0.88"	"0.95"	"0.92"	"0.87"	"0.93"
"40"	"0.85"	"0.91"	"0.84"	"0.76"	"0.93"
"30"	"0.81"	"0.87"	"0.67"	"0.8"	"0.82"
"20"	"0.72"	"0.75"	"0.51"	"0.59"	"0.78"
"10"	"0.49"	"0.47"	"0.32"	"0.4"	"0.43"
"5"	"0.51"	"0.35"	"0.19"	"0.28"	"0.25"

Figure 4.1 The mean of effectivity of all languages results going from good to better: Unigrams $\square$ and Bigrams $\triangle\times$



Programming languages and graphics programs made with GNUplot, R, PovRay and Perl (Persistence of Vision Pty. Ltd. 2004; R Core Team 2021; Wall and The Perl Foundation 2024; Williams et al. 2024).

5 Conclusions

The findings testing 5500 texts in this study indicate the following:

All languages got superior to 18% ($P \geqslant 0.19$) detecting rate from text lengths of 5 or more characters for drawn bigrams, getting (Table 4.2) or superior to 15% in texts lengths of 5 to draw unigrams but italian, that needed at least 10 character segments to get the phoneme unigrams (Table 4.1).

Hypothesis H_0 is "no language detection" and as there are 10 different languages baricenters where to choose from 100 takes, so 0.1 is the probability of any of them when a uniform distribution random take (6)

$$P(H_0) = P(X > 15) = 1 - \sum_{i=0}^{15} \binom{100}{i} (0.1)^i (0.9)^{100-i} = 1 - pbinom(15, 100, 0.1) = 4\% \quad (7)$$

therefore $H_1 \Rightarrow$ "the class language is detected" with a superior confidence to 95% in all the tests but Italian in 5 character segments when using unigrams, and detected for any case from 10 characters on, and 100% effectivity for all the languages when using 100 or more simplified characters (phonemes) texts segments with drawn Bigrams (Table 4.2).

Character Bigrams the most effective representation for language classification tasks compared with character Unigrams. The precision can be adjusted by adjusting the n-gram size.

The method achieved strong classification performance even with very short text fragments (as few as five characters).

Compared to the widely used approach by Cavnar & Trenkle, which reports lower accuracy for Spanish and Italian in small text samples, the proposed method demonstrated superior performance with fragment lengths of 100 characters or fewer.

Encoding text as points in real Euclidean space provides comparable, if not superior, results to traditional discreet vector-based models while offering computational advantages in terms of complexity, computation power and Carbon footprint.

Additionally, the neutrality of this statistical approach—requiring no manual feature preselection—suggests its applicability to broader classification tasks, including authorship attribution and linguistic evolution studies.

The computed language centroids and the code framework will be made available upon request or hosted in a public repository upon publication.

6 Declarations

6.0.1 Funding and/or Conflicts of interests/Competing interests.

The authors declare that they have no competing interests.

References

Basile C, Benedetto D, Caglioti E, et al (2008) An example of mathematical authorship atribution. Journal of Mathematical Phisics

BibleGateway.com (2025) Biblegateway.com - over 50 online bibles in 35 languages, in text and audio format. URL http://www.biblegateway.com/versions/

Brown RD (2013) Selecting and weighting n-grams to identify 1100 languages. Lecture Notes in Computer Science https://doi.org/10.1007/978-3-642-40585-3_60

Cavnar WB, Trenkle JM (1994) N-gram based text categorization. In: Proceedings of 3rd. Annual Symposium on Document Analysis an Information Retrieval

Clement R, Sharp D (2003) Ngram and bayesian classification of documents for topic and authorship. Literary and Linguistic Computing 18(4):423–447. URL http://dx.doi.org/10.1093/llc/18.4.423

Damashek M (1995) Gauging similarity with n-grams: Language-independent categorization of text. Science 267(5199):843–848. https://doi.org/10.1126/science.267.5199.843, URL http://www.sciencemag.org/content/267/5199/843.full.pdf

Dubin D (2004) The most influential paper gerard salton never wrote. Library Trends 52:748–764

Dunning T (1994) Statistical identification of language. Tech. rep., New Mexico State University

Jadalla A, Elnagar A (2007) Pde4java: Plagiarism detection engine for java, source code: A clustering approach. In: Proceedings of iiWAS2007, pp 147–155

Keselj V, Peng F, Cercone N, et al (2003) N-gram-based author profiles for authorship attribution. Proceedings of the Conference Pacific Association for Computational Linguistics PACLING'03: 2003

Koehn P (2005a) Europarl: A parallel corpus for statistical machine translation. In: Proceedings of the Tenth Machine Translation Summit, Phuket, Thailand, pp 79–86, URL https://aclanthology.org

Koehn P (2005b) Europarl: A parallel corpus for statistical machine translation. In: MT Summit, URL http://www.statmt.org/europarl/

Lukashenko R, Graudina V, Grundspenkis J (2007) Computer-based plagiarism detection methods and tools: An overview. In: International Conference on Computer Systems and Technologies, pp IIIA.18–1–6

Persistence of Vision Pty. Ltd. (2004) Persistence of Vision Raytracer (Version 3.6). http://www.povray.org/download/, [Computer software]

R Core Team (2021) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, URL https://www.R-project.org/

Salton G, Wong A, Yang CS (1975) A vector space model for automatic indexing. Commun ACM 18:613–620. URL http://doi.acm.org/10.1145/361219.361220

Shannon CE (1948) A mathematical theory of communication. The Bell System Technical Journal 27:379–423

Tan CL, Sung SY, Yu Z, et al (2000) Text retrieval from document images based on n-gram algorithm. In: Text and Web Mining Workshop, 6th Pacific Rim International Conference on Artificial Intelligence, Publisher, pp 257–270

Tommi Jauhiainen KLHeidi Jauhiainen (2022) Heli-ots, off-the-shelf language identifier for text. In: Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), p 3912–3922

Wall L, The Perl Foundation (2024) The perl programming language. URL http://www.perl.org/

Williams T, Kelley C, many others (2024) Gnuplot 6.0: an interactive plotting program. http://gnuplot.info

Wong SKM, Ziarko W, Wong PCN (1985) Generalized vector spaces model in information retrieval. In: Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, New York, NY, USA, SIGIR '85, pp 18–25, URL http://doi.acm.org/10.1145/253495.253506

Yannakoudakis E, Goyal P, Huggill J (1982) The generation and use of text fragments for data compression. Information Processing and Management 18(1):15–21. https://doi.org/https://doi.org/10.1016/0306-4573(82)90047-4, URL https://www.sciencedirect.com/science/article/pii/0306457382900474

Zamora EM, Pollock JJ, Zamora A (1981) The use of trigram analysis for spelling error detection. In: Information Processing and Management, pp 305–316